Claims

1. Method of storing data in a random access memory in which data words, which each comprise a predetermined number of data bits, are storable, *characterized in* that, before storage, an encryption of each data word (M) is effected whereby a permutated data word (P) with a predetermined number of data bits is generated from each data word (M), or from a data word (M) derived from this data word, by one-to-one permutation of the individual data bits (M[n-1]-M[0]) using a first permutation key (Mp).

2. Method according to Claim 1, in which the individual data bits (M[n-1]-M[0]) of the permutated data word (Mp) are substituted before storage using a first substitution key in order to provide an encrypted data word (M').

3. Method according to Claim 1, in which before rearrangement the individual data bits of the data word (M) are substituted using a first substitution key (S) in order to provide a substituted data word.

4. Method according to one of the foregoing claims, in which the permutation key (P) has a number of unique subkeys (P[n-1]-P[0]) corresponding to the number n data bits, which subkeys are assigned to one data bit each (Mp[n-1]-Mp[0]) of the permutated data word (Mp), and which each indicate the data bit (M[n-1]-M[0]) of the data word to be permutated (M), which data bit is to be mapped to this data bit (Mp[n-1]-Mp[0]), wherein each subkey (P[n-1]-P[0]) comprises a number of key bits (P[n-1,m-1]-P[n-1,0], P[k,m-1]-P[k,0], P[0,m-1]-P[0,0]).

5. Method according to Claim 4, in which the mapping of a data bit (M[n-1]-M[0]) of the data word to be permutated (M) to a data bit (Mp[k]) of the permutated data word is effected incrementally using a subkey (P[k]) by the following steps:

a) selecting a first group of data bits of the data word to be permutated (Mp) as determined by a first key bit (P[k,0]) of the subkey (P[k]);

b) selecting a second group of data bits from the first group of data bits obtained by the previous selection as determined by a second key bit (P[k,1]) of the subkey (P[k]);

c) repeating step b), each time using an additional key bit (P[k,2]...P[k,m-1]) until the selected group comprises only one more data bit which corresponds to the data bit (Mp[k]) of the permutated data word (Mp).

6. Method according to Claim 5, in which the number of data bits contained in a group of data bits is reduced from one step to the next by a factor of 2.

7. Method according to one of the foregoing claims, in which the first substitution key (S) has a number of key bits (S[n-1]...S[0]) corresponding to the number of data bits of the data word to be substituted (Mp), wherein each data bit of the data word to be substituted (Mp) is mapped unchanged or inverted to a data bit (M'[n-1]...M'[0]) of the substituted data word (M') as determined by one of these key bits (S[n-1]...S[0]).

8. Method according to one of the foregoing claims, in which the permutation key (P) and the substitution key (S) are regenerated before a rewriting to the memory after a deletion.

9. Method according to one of the foregoing claims, which in order to generate a permutation key (P) comprises the following steps:

a) randomly generating a sub-permutation-key and assigning the subkey to a bit position of the permutated data word;

b) checking whether the generated sub-permutation-key has already been generated for another bit position of the permutated data word, and retaining the generated sub-permutation-key if it has not yet been generated, and rejecting the generated sub-permutation-key if it has already been generated;

c) implementing steps a) and b) until a subkey is generated for each bit position of the permutated data word (Mp).

10. Method according to one of the foregoing claims, in which a data word (M'), generated from a data word (M) using the first key after being read out from the memory, is permutated in order to generate the data word using a second permutation key (P') which is matched to the first permutation key (P).

11. Device to encrypt/decrypt a data word (M) comprising data bits (M[n-1], M[k], M[0]), which device has a permutation unit (14) with the following features:

- data inputs to supply the data bits (M[n-1], M[k], M[0]) of the data word to be permutated (M);

- outputs to supply the data bits (Mp[n-1], Mp[k], Mp[0]) of a permutated data word (Mp) of the predetermined length (n);

- permutation key inputs to supply a permutation key (P) which comprises a number (n) of subkeys (P[n-1]... P[0]) corresponding to the number of data bits;

- a number of selection units (14_n-1, 14_k, 14_0) corresponding to the number of data bits, to which selection units one subkey each is assigned and which provide one data bit each (Mp[n-1], Mp[k], Mp[0]) of the permutated data word (Mp) as determined by one each of the subkeys (P[n-1]... P[0]) from the data bits of the data word to be permutated (M).

12. Device according to Claim 11, in which each of the selection units (14_k) has a number of consecutively arranged selection stages (141_n-1, 141_k, 141_0) corresponding to the number of permutation key bits, wherein a first selection stage (141_0) is designed, as determined by a first key bit (P[k,0]), to select and provide a first group of data bits from the data word to be permutated (M), and wherein subsequent selection stages (141_1, 141_2, 141_m-1) are designed, in each case as determined by a key bit (P[k,1], P[k,2], P[k,m-1]), to select a subgroup from the group of data bits provided by the respective previous selection stage.

13. Device according to Claims 11 or 13, in which a substitution unit is connected before or after the permutation unit (14), which substitution unit substitutes data bits (Mp[n-1], Mp[k], Mp[0]) of a data word to be substituted (Mp) as determined by a substitution key (S).